AoM33PIS - Průmyslové informační systémy

Přednáška č. 9

20. 4. 2016

Katedra Kybernetiky K13133
Centrum znalostního managementu K13393

Víme co a proč chceme?

„Všechny šťastné rodiny jsou si podobné; každá nešťastná rodina je nešťastná po svém.“

L. N. Tolstoj, Anna Karenina

Agenda

- Spolehlivost a dostupnost.
- Kvalita (jakost) v informatice a informačních systémech.
- Základy metodologií.

SPOLEHLIVOST A DOSTUPNOST

Spolehlivost a dostupnost systému

- **Porucha -** *dvě* základní definice poruchy:
  1. Ukončení schopnosti produktu jako celku vykonávat požadovanou funkci.
  2. Ukončení schopnosti libovolné součásti vykonávat požadovanou funkci, aniž by musel selhat celý produkt.
  - *Příklad:* Pokud dojde k selhání redundantního disku v poli RAID, bude diskové pole RAID nadále fungovat a poskytovat kritická data. Selhání disku však způsobí, že součást diskového pole nebude vykonávat požadovanou funkci, tj. poskytování úložného místa. Podle definice 1 se tedy nejedná o poruchu, ale podle definice 2 se o poruchu jedná.
- **Spolehlivost** je schopnost systému nebo součásti vykonávat požadované funkce za daných podmínek po určené časové období.
- **Dostupnost** na druhé straně představuje úroveň, do které je systém nebo součást funkční a k dispozici v případě, že je vyžádáno její použití. Dostupnost lze považovat za pravděpodobnost, že se systém nebo součást nachází ve stavu, kdy umožňuje provádět požadované funkce za určených podmínek a v daném časovém okamžiku.
- Dostupnost je určena spolehlivostí systému a časem obnovení v případě poruchy.

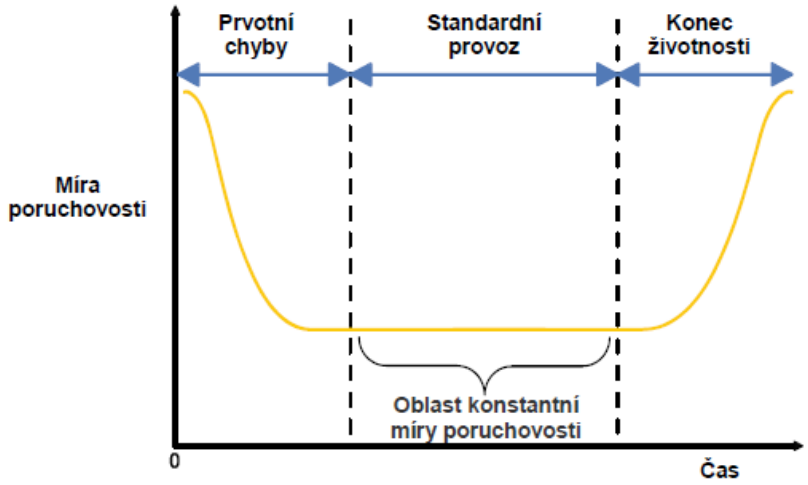
$$Dostupnost = \frac{MTBF}{MTBF + MTTR}$$

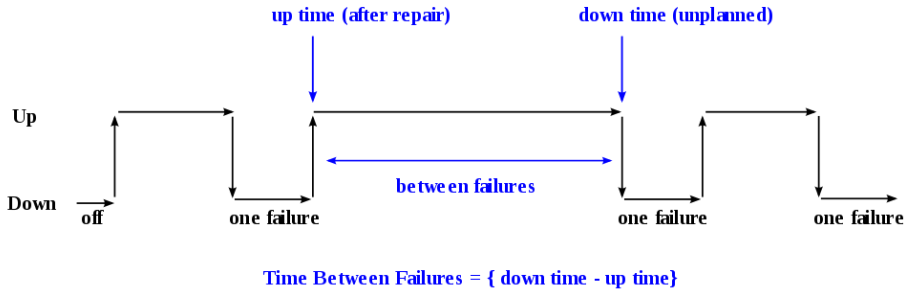
- ***Příklad:*** *dostupnost 99,99% pro 24x7x365: celkem 8760, TTR = 0,876 hod.*

http://www.apcmedia.com/salestools/VAVR-5WGTSB_R0_CZ.pdf

MTBF, MTTF

- **Střední doba mezi poruchami** (MTBF, *Mean Time Between Failures*) - statistická veličina, sloužící k ohodnocení spolehlivosti systému, u kterého se předpokládá okamžitá oprava.
- Střední doba do poruchy (MTTF, *Mean Time to Failure*) – pro zařízení, která se neopravují.
- U komplexních systémů zohledňuje statistické ohodnocení poruchovosti jednotlivých komponentů. Pro elektronické systémy se obvykle předpokládá, že komponenty mají exponenciální rozdělení pravděpodobnosti poruchy a MTBF je konstantní po dobu standardního provozu systému





MTBF, MTTF

- MTBF lze počítat takto:

$$\mathrm{MTBF} = \frac{\Sigma(\mathrm{downtime} - \mathrm{uptime})}{\mathrm{number\ of\ failures}}.$$

a pravděpodobnost, že systém bude pracovat bez poruchy po dobu T (spolehlivost systému):

$$R(T) = e^{-\frac{T}{MTBF}}$$

Příklad: *Systém s MTBF 250.000 hod., plánovaná doba nepřetržitého provozu 5 let (43.800 hod):*

$$R(T) = e^{-\frac{43800}{250000}} = 0{,}839$$

tj. je pravděpodobnost 83.9%, že systém bude pracovat 5 let bez poruchy (respektive, že 83,9% z provozovaných systémů bude po 5 letech stále pracovat).

MTBF, MTTF

- MTBF je často chybně interpretována jako předpokládaný počet provozních hodin před selháním systému nebo jako „servisní životnost“.
- MTBF jsou založeny na pravděpodobnosti poruch produktu při „běžných podmínkách“ nebo „při standardním provozu“ a předpokládá se, že pravděpodobnost poruchy se s časem nemění a je stejná bez ohledu na dobu provozu. V této fázi životnosti produktu se dosahuje nejnižší (a konstantní) pravděpodobnosti poruchy.
- Provoz systému omezuje doba jeho životnosti, která je podstatně kratší než hodnoty MTBF. Je docela možné vyrobit produkt s extrémně vysokou spolehlivostí (MTBF), který však bude mít krátkou očekávanou životnost.
- ***Příklad:*** *uhlíková baterie může mít za daných podmínek životnost 4 hod. a MTBF 100.000 hod. To lze interpretovat tak, že v množině 1.000.000 baterií se vyskytne během jejich čtyřhodinové životnosti u 10 ks každou hodinu porucha*
- Další používané odvozené charakteristiky:
  - MTBSA - **mean time between system aborts**
  - MTBCF - **mean time between critical failures**
  - MTBUR - **mean time between unit replacement**.

MDT, MTTR

- **Střední doba výpadku** (MDT, mean down time) - střední doba, po kterou je systém mimo provoz. Zahrnuje veškeré časy opravy, preventivní údržby, odstávky aj.
- **Střední doba opravy (obnovy)** (MTTR, Mean Time to Repair) - očekávaný časový interval, během kterého dojde k obnovení systému po poruše. Zahrnuje čas pro diagnostiku a celkovou dobu opravy systému.
- MTTR je obvykle součástí servisní smlouvy na údržbu IS - „měkká“ podmínka, negarantuje absolutní čas, ale průměrnou trendovou hodnotu. Vhodnější je použít charakteristiku „maximální doba opravy“. Někteří dodavatelé interpretují MTTR jako „mean time to respond“, tj. reakční doba bez garance odstranění poruchy.
- Charakteristiky *MDT* a *MTTR* lze exaktně stanovit.
- Charakteristika *MTBF* se obvykle odhaduje na základě sledování vzorku podobných systémů, který je obvykle analyzován po implementaci dostatečně velkého počtu produktů do provozu.

Fault Tolerant (FT) systémy

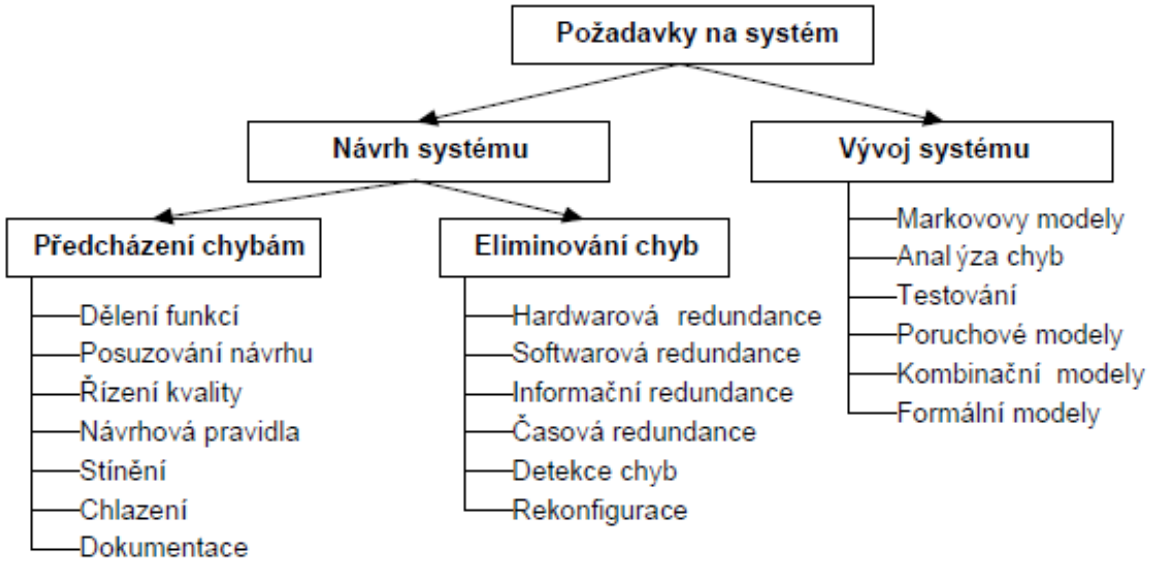
- Příklad: OES (Operations Execution System) řídí na výrobní lince otevírání/zavírání ventilu. Polohovací ventil se při log.1 na řídící sběrnici zavírá, log.0 otevírá.
  - Chyba: neošetřená výjimka v SW, zkrat na sběrnici, aj. – trvale log.0
  - Porucha: systém otevírá ventil
  - Selhání: vodárna přeteče
  - Havárie: zaplavená výrobní hala

- ... systém nebyl navržen s ohledem na správnou reakci na tuto chybu - systém není Fault Tolerant

Kučera: Fault Tolerant Systémy. CAK VUT Brno, http://sciotech.cz/tc/mrts.php

Metodika FT návrhu

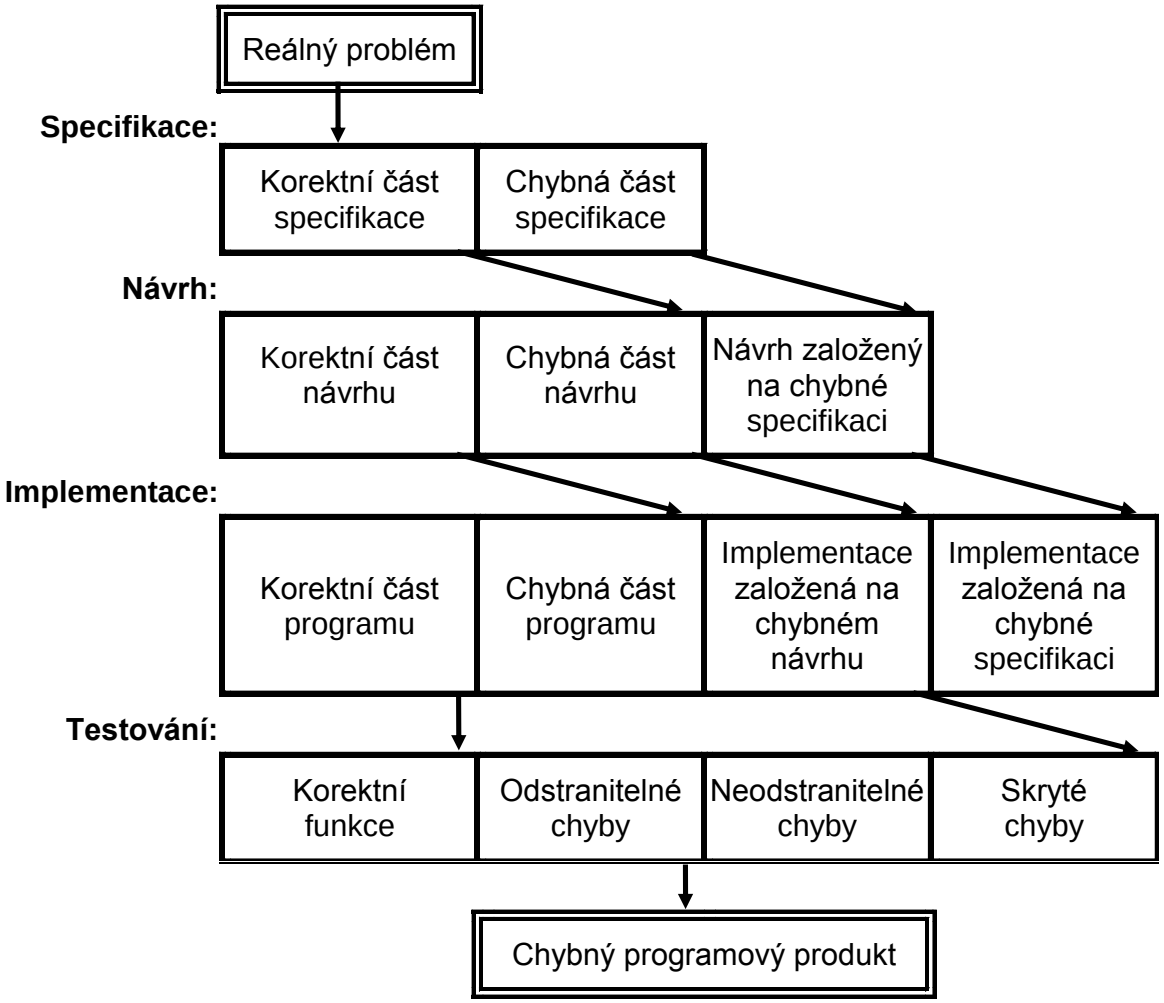
- základní postupy při návrhu FT systémů, kterými eliminujeme (minimalizujeme) vliv chyb na systém



- Použití jak pro hardwarovou, tak i pro softwarou část řešení
- **Softwarová redundance** – realizace stejného algoritmu různými dodavateli, v odlišném programovacím jazyce, odlišném vývojovém prostředí, pro odlišný operační systém

Kumulace SW chyb při návrhu IS

Vliv chyb při vývoji softwaru na programový produkt



Spolehlivostní modely

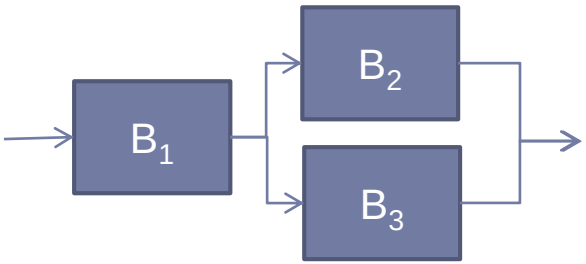
Spolehlivostní modely

- predikce spolehlivosti zejména při návrhu systémů pro kritické aplikace
- ukazatele spolehlivosti odvozovány z informací o jednotlivých komponentech (blocích) a způsobu jejich použití
- existuje řada metod, všechny jsou pracné, zjednodušující

Blokové spolehlivostní modely (Reliability Block Model, RBM)

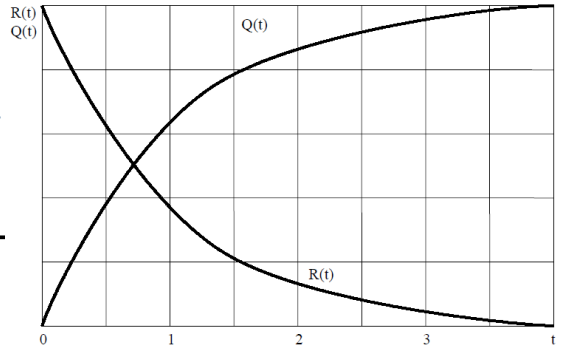
- každá komponenta reprezentována blokem, každý blok popsán spolehlivostními parametry
- komponenty jsou vzájemně nezávislé (z hlediska výskytu poruchy)
- RBM je orientovaný graf, hrany tvoří orientovanou cestu mezi vstupem a výstupem, každá cesta popisuje jeden provozuschopný stav systému
  - systém je bezporuchový, jsou-li bezporuchové všechny prvky ležící na alespoň jedné cestě, spojující vstup a výstup

Příklad



Systém je provozuschopný, je-li současně B_1 a B_2 nebo B_1 a B_3 v bezporuchovém stavu

Spolehlivostní modely



- Intenzita poruch λ_k jedné komponenty *k* [čas^{-1}]:

$$\lambda_k = \frac{1}{MTBF}$$

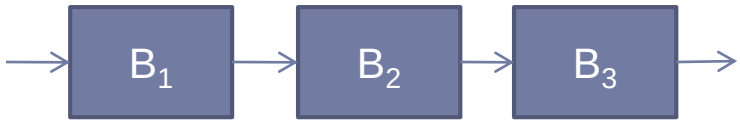
- Pravděpodobnost bezporuchového provozu jedné komponenty *k* s intenzitou poruch λ_k po dobu *t*:

$$R(t) = e^{-\lambda_k t}$$

Vhodná aproximace v období *standardního* provozu

Sériový model

- Sériový model - porucha kterékoliv komponenty systému způsobí poruchu v celém systému.



- Známe-li pravděpodobnost bezporuchového provozu každé z komponent, pak výsledná pravděpodobnost bezporuchového provozu:

$$R_S(t) = \prod_{i=1}^{n} R_i(t)$$

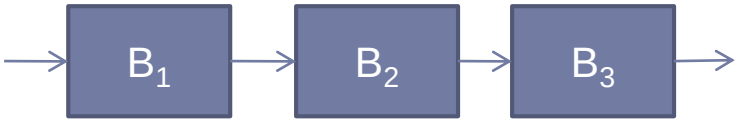
- Má-li každá komponenta intenzitu poruch λ_i , pak výsledná intenzita poruch systému:

$$R_S(t) = \prod_{i=1}^{n} e^{-\lambda_i t} = e^{-\lambda_s t}$$

kde $\lambda_s = \sum_{i=1}^{n} \lambda_i$

Spolehlivostní modely

- MTBF, střední doba bezporuchového provozu sériového systému:



$$MTBF = \frac{1}{\lambda_s} = \frac{1}{\sum_{i=1}^{n} \lambda_i}$$

- Mají-li všechny komponenty shodné intenzity poruch λ, pak

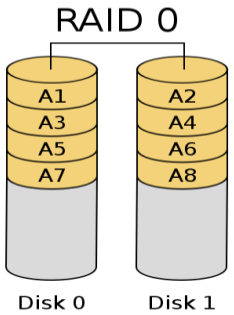
$$R_S(t) = \prod_{i=1}^{n} e^{-\lambda_i t} = e^{-n\lambda t}$$

- a střední doba bezporuchového provozu jednoduše $MTBF_s = \frac{1}{n\lambda}$

Příklad:

Diskové pole RAID 0 s prokládáním dat. Použité disky mají MTBF 400.000 hod.

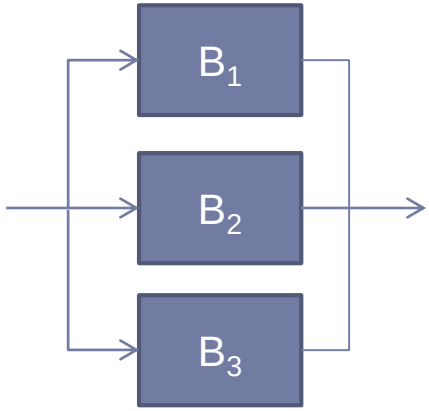
Pravděpodobnost bezporuchového provozu po třech letech?



$$R_S(t) = e^{-n\lambda t} = e^{-2\frac{1}{4.10^5}.3.365.24} = e^{-0,1314} = 0.877$$

Spolehlivostní modely

Paralelní model



- Paralelní model - porucha celého systému nastane, dojde-li k poruše všech komponent systému.
- Výsledná pravděpodobnost poruchy je součin pravděpododbností poruch všech komponent (nezávislé komponenty):

$$Q_p(t) = \prod_{i=1}^{n} Q_i(t)$$

- a pravděpodobnost bezporuchového provozu paralelního systému v čase *t:*

$$R_p(t) = 1 - Q_p(t) = 1 - \prod_{i=1}^{n} Q_i(t) = 1 - \prod_{i=1}^{n} (1 - R_i(t)) = 1 - \prod_{i=1}^{n} (1 - e^{-\lambda_i t})$$

- MTBF, střední doba bezporuchového provozu paralelního systému s *n* stejnými komponentami s intenzitou poruch λ:

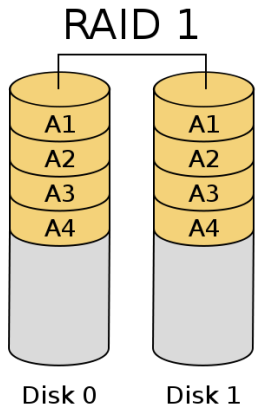
$$MTBF_p = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{1}{i}$$

Spolehlivostní modely

Příklad

Diskové pole RAID 1 se zrcadlením dat.
Použité disky mají MTBF 400.000 hod.

Pravděpodobnost bezporuchového provozu po třech letech?



$$R_p(t) = 1 - \prod_{i=1}^{n} (1 - e^{-\lambda_i t})$$

$$R_p(t) = 1 - (1 - e^{-\frac{1}{4.10^5}.3.365.24})(1 - e^{-\frac{1}{4.10^5}.3.365.24}) = 1 - (1 - e^{-0,0657})^2 = 0,99596$$

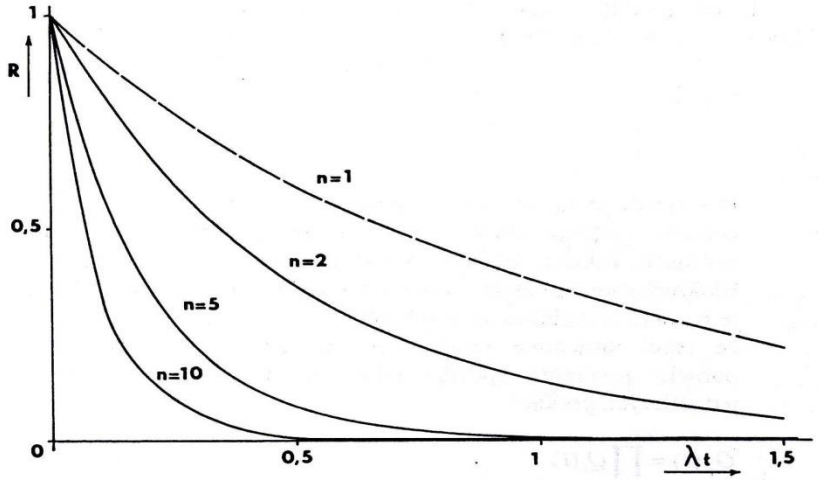
Pro srovnání, pravděpodobnost bezporuchového provozu samotného disku po třech letech:

$$R(t) = e^{-\lambda t} = e^{-\frac{1}{4.10^5}.3.365.24} = e^{-0,0657} = 0,9364$$

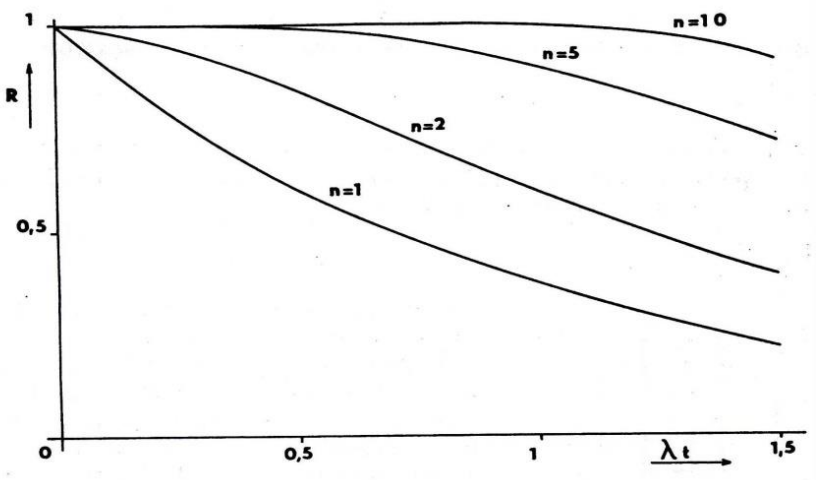
a pro RAID 0 jsme spočítali 0,877

Spolehlivostní modely

- Srovnání charakteristik seriového a paralelního modelu



Pravděpodobnost bezporuchového provozu **sériového systému** s *n* bloky v závislosti na intenzitě poruch



Pravděpodobnost bezporuchového provozu **paralelního systému** s *n* bloky v závislosti na intenzitě poruch

- **Sériové modely** jsou velmi **časté**, ale čistě **paralelní modely** spolehlivosti **jsou velmi ojedinělé** (avšak záměrně realizované)
- V praxi jsou nejčastější tzv. **kombinované modely**, v nichž se vyskytují různé kombinace sériových a paralelních systémů.
- K řešení kombinovaných modelů spolehlivosti můžeme přistupovat jako k řešení **paralelního uspořádání sériových** nebo **sériového uspořádání paralelních** dílčích modelů.

Spolehlivostní modely

Systémy M z N

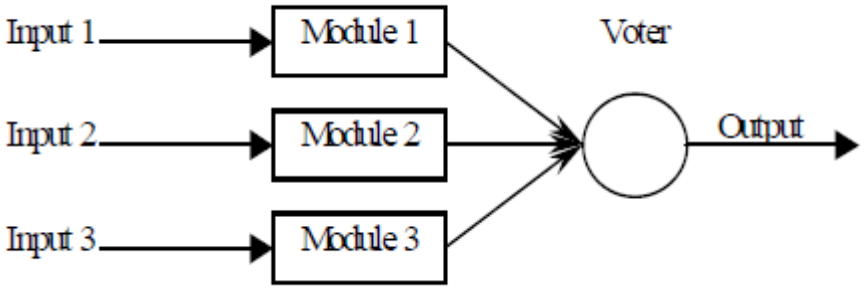
- Jsou generalizací ideálních paralelních systému. V M z N systému je nutné ke správné činnosti systému jeho **M prvku z** celkových **N prvku.**
- Pravděpodobnost bezporuchového provozu systému M z N:

$$R_{MzN}(t) = \sum_{i=0}^{N-M} \binom{N}{i} R^{N-i(t)(1-R(t))^i}$$

DMR a TMR systémy

- **DMR** (Dual Modular Redundand) pouze zdvojení
- **TMR** (Triple Modular Redundand) uspořádání tří prvků tak, aby výpadek jednoho vedl k maskování poruchy v systému

TMR model podle J. von Neumanna, 1956



$$R_{TMR}=3R^2 - 2R^3$$

$$R_{TMR}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}$$

Spolehlivostní modely

- Výsledná spolehlivost IS je určena současně:
  - hardwarovou spolehlivostí, tj. spolehlivostí technické infrastruktury,
  - softwarovou spolehlivostí, tj. spolehlivostí algoritmizace a softwarové realizace,
  - informační spolehlivostí, tj. spolehlivostí přenosu, zpracování a uchovávání informací,
  - spolehlivostí lidského činitele.
- Cílem je zabezpečit odolnost proti vytipovaným poruchám s nejkritičtějšími následky, což se označuje jako koncepce bezpečnosti při poruše (návrhová vlastnost systému, která zabraňuje, aby jeho poruchy vedly ke kritickým poruchovým stavům):
- použitím redundantního technického vybavení, tzv. hardwarového zálohování,
- použitím softwarové redundance s využíváním návrhu programů s ohledem na spolehlivost, tj.
  - začleněním kontrolních funkcí, resp. algoritmů diagnostiky,
  - při programové realizaci využíváním jednoduchých programových struktur s vhodnou volbou kontrolních bodů,
  - využíváním analytických metod testování a verifikací navržených programů;
  - opakování téhož výpočtu podle různých algoritmů
  - využívání nadbytečnost informační - použití např. redundantního kódování

Spolehlivostní modely

- Z toho pro dosahování vyšší bezporuchovosti **sériového systému** vyplývá:
  - Minimalizace počtu prvků při zvolené úrovni rozkladu technické a algoritmické struktury systému na prvky (tj. bloky), tj. volba rozsahu a kvality funkčních a dalších užitných vlastností systému, které právě splňují požadavky zákazníků.
  - Zvyšováním bezporuchovosti prvků systému, tj. zvýšením hodnot $R_i(t)$, resp. snížením hodnot l_i. Bezporuchovost použitých technických prostředků je dána nejen jejich inherentní bezporuchovostí, odrážející použité výrobní technologie, ale rovněž způsobem využití v systémech (volbou pracovních režimů, zatížení aj.).
- Technická realizovatelnost principů **paralelního systému** - hardwarového zálohování - vede na substituční (dynamické) zálohování, kdy záložní prvek přebírá funkci zálohovaného prvku teprve po detekci poruchy podle pracovního režimu:
  - zatížené SZ - základní i záložní prvek začnou pracovat současně po uvedení systému do provozu, na všechny prvky působí stejná provozní zatížení,
  - odlehčené SZ, kdy v plném pracovním režimu je pouze základní prvek systému a záložní prvek je v odlehčeném pracovním režimu (tj. např. spuštěn, ale bez provozního zatížení),
  - nezatížené SZ, kdy je základní prvek v plném pracovním režimu, prvek zálohy je mimo provoz; po poruše základního prvku je prvek z nezatížené zálohy převáděn do plného pracovního režimu.
- Přepnutí na záložní prvek je doprovázeno krátkodobým narušením provozuschopnosti systému, které buď z hlediska provozu systému nemá význam a nehodnotí se jako jeho porucha, nebo musí být ošetřeno, např. odpovídajícím programovým opatřením.

Spolehlivostní modely

Kromě uvedených základních modelů zálohování se v praxi využívá mnoho dalších, složitějších, např.:

- **Substituční zálohování s pohyblivou zálohou** – skupina stejných prvků má jeden nebo několik záložních prvků a v případě poruchy kteréhokoliv z prvků základní skupiny je na jeho místo připojen záložní prvek (resp. jeden ze skupiny záložních prvků).
- **Zálohování se sdílením zátěže** – při normálním pracovním režimu jsou realizované funkce, resp. zatížení rozloženy např. na dva prvky, které pracují v částečně odlehčeném režimu a jsou dimenzovány tak, že v případě poruchy jednoho z nich přebírá druhý prvek všechny funkce, resp. celé zatížení, a pracuje pak s nominálním, resp. maximálním zatížením, zpravidla navíc při určitém (nevýznamném) omezení rozsahu funkcí systému.
- **Zálohování s obecnější rekonfigurací struktury** – dynamické zálohování s obnovou prvků po poruše a s rekonfigurací poruchových modelů. Např. z výchozího majoritního zálohování dva ze tří se po poruše libovolného prvku přechází na paralelní systém se dvěma prvky a po eventuální další poruše zbývá neporouchaný pouze základní prvek. Struktura bývá rekonfigurována automaticky s využitím tzv. diagnostického procesoru, na který navazuje činnost rekonfigurační jednotky.

Příklad – komponenta palubního počítače

- Komponenta palubního počítače letounu Grumman X-29 pro řízení letu je tvořena:
  - 3 snímače polohy s intenzitou poruchy λ_s,
  - 3 snímače povelů pilota s intenzitou poruchy λ_p,
  - 3 akční členy s intenzitou poruchy λ_a,
  - 3 mikropočítače s intenzitou poruchy λ_m,
  - řídící sběrnicí s intenzitou poruch λ_{bc}
  - datovou sběrnicí s intenzitu poruch λ_{bd}.
- Jaká je pravděpodobnost bezporuchového provozu systému po dobu 5 hodin letu (selhání způsobí výpadek libovolného prvku).
- Sériový systém – intenzita poruch:

$$\lambda_{system} = 3\lambda_s + 3\lambda_p + 3\lambda_a + 3\lambda_m + \lambda_{bc} + \lambda_{bd} \qquad (s^{-1})$$

- Pravděpodobnost, že systém bude pracovat správně:

$$R(18000) = e^{-\lambda_{system} 18000}$$

http://www.nasa.gov/centers/dryden/news/FactSheets/FS-008-DFRC.html

Příklad – komponenta palubního počítače

Parametry letounu X-29:

- $\lambda_s = 10^{-6}\ h^{-1}$
- $\lambda_p = 10^{-6}\ h^{-1}$
- $\lambda_a = 10^{-5}\ h^{-1}$
- $\lambda_m = 4 .10^{-4}\ h^{-1}$
- $\lambda_{bc} = 10^{-6}\ h^{-1}$
- $\lambda_{bd} = 2 .10^{-6}\ h^{-1}$

- Po dosazení – intenzita poruch:

  $\lambda_{system} = 1{,}239 \times 10^{-3}\ h^{-1}$

- a pravděpodobnost bezporuchového provozu po dobu 5-ti hodin:

  $R(5\ h) = 0{,}995$

(letěli byste s tím?)

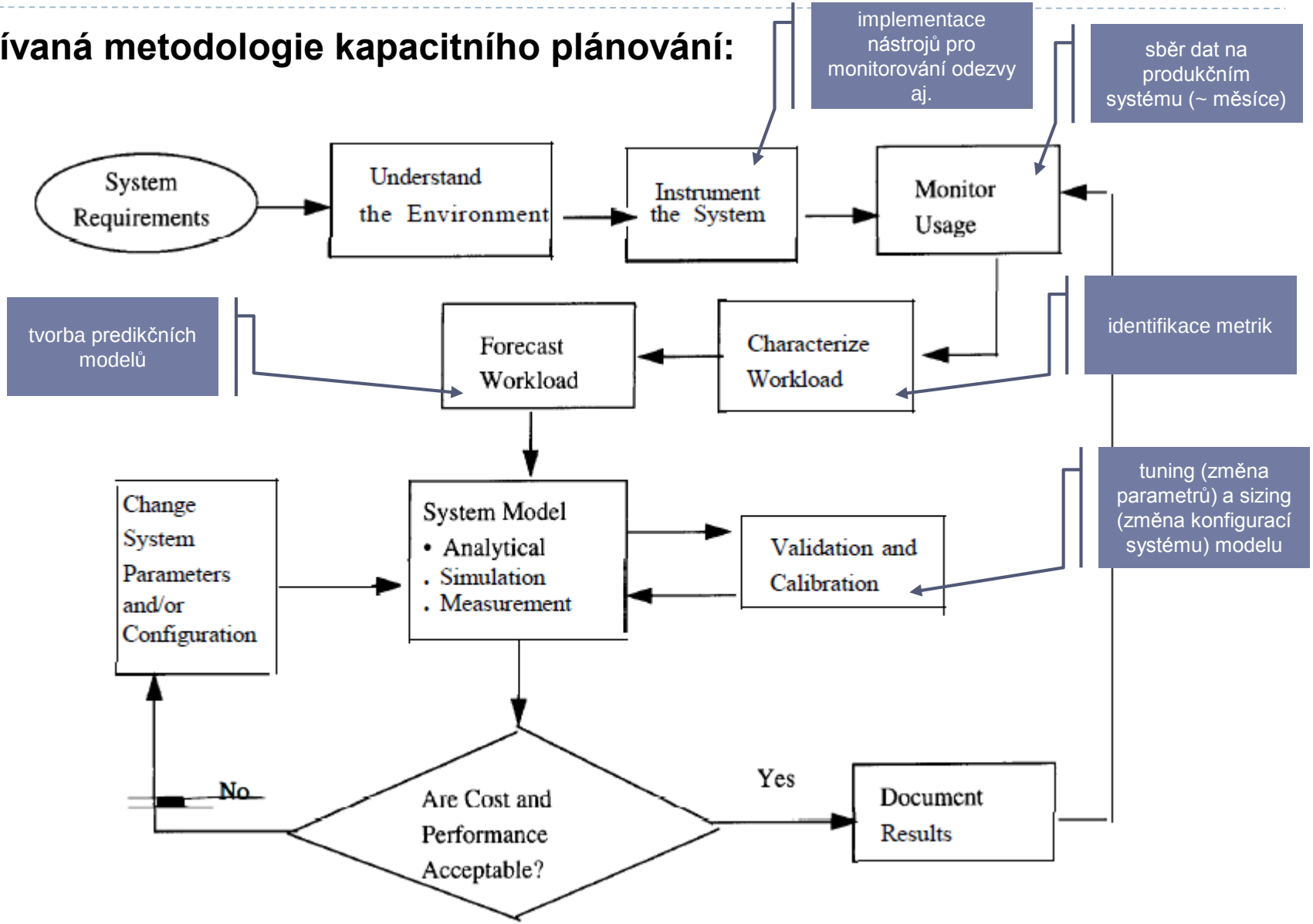
Kapacitní plánování systému

Základní otázky pro kapacitní plánování IS:

- Jaké výkonostní metriky (data) mají být sledovány?
- Jak často mají být data sbírána?
- Co jsou relevantní prahové úrovně nebo akceptovatelné provozní úrovně?
- Co se má provést, jsou-li určité prahové nebo provozní úrovně překročeny?
- Jak je sbírána statistika pro charakterizaci zatížení, jeho predikce, modelování výkonnosti, kapacitní plánování a konfiguraci?
- Kdo jsou účastnící těchto procesů? a
- Jaké jsou jejich role?

Kapacitní plánování systému

Používaná metodologie kapacitního plánování:

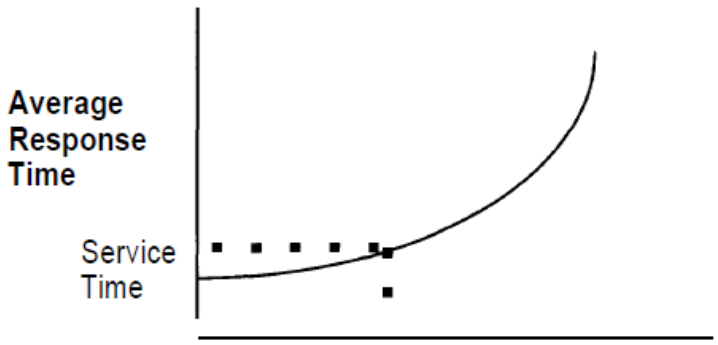


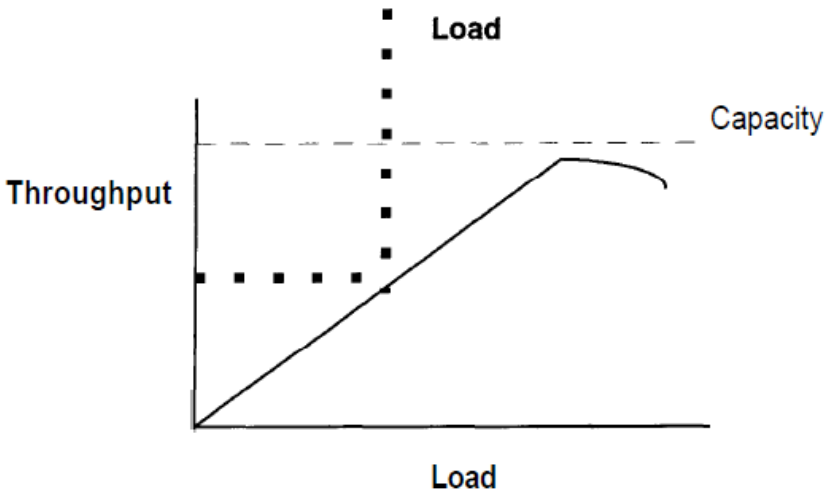
Kapacitní plánování systému

Příklad tvorby performance modelu:

typické sledované parametry:

- propustnost systému
- kapacita systému
- doba odezvy





typicky sbíraná data pro jednotlivé transakce:

- start/stop čas
- vytížení sítě
- obsazení paměti
- strojový čas každého z procesorů
- vytížení každého z procesorů (%usr+%sys)
- transakční statistika – typ transakce, uživatelská skupina, odezva aj.

KVALITA V INFORMATICE

Inspirujme se tím, co je nám blízké

- Optimization is a structured, systematic process of assessing maturity across IT capabilities, then prioritizing projects to progress towards a Dynamic state.

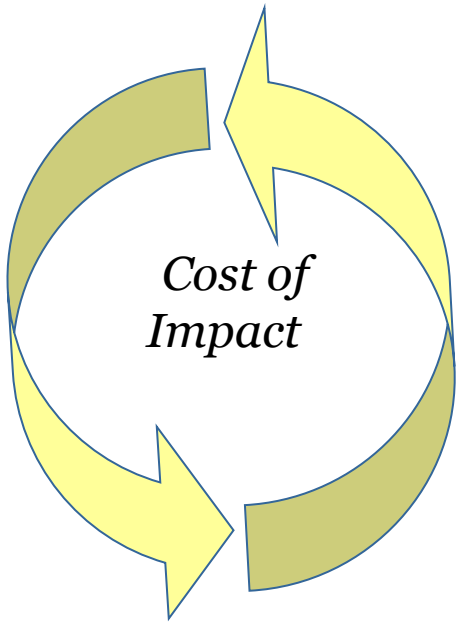
http://www.microsoft.com/optimization/default.mspx

Business Impact Matrix (BIM)

Lost Revenue
- *Billing Loss on 24/7 projects*
- *Lost Future Revenues*
- *Compensatory Payments*
- *Suboptimal Investment Utilization*

Productivity Loss
- *Lost man hours*
- *Compensatory Overtime*
- *Lost in progress data*
- *Lost momentum*

Damaged Reputation
- *Customer, Suppliers, Partners. Banks, Financial Markets*
- *Employee morale*
- *Credit Rating*
- *Customer Confidence*



Extra Expense
- *Cost to Recover*
- *Increased Fraud Risk*
- *Increased Error Rate*
- *Travel Expenses*
- *Temporary Employees*

Legal & regulatory liabilities
- *Contractual Penalties per contract clause*
- *Regulatory Issues*
- *Legal Issues*

Delayed Collections
- *Billing Losses*
- *Missed documents*
- *Report Outs*

K čemu lze využít BIM?

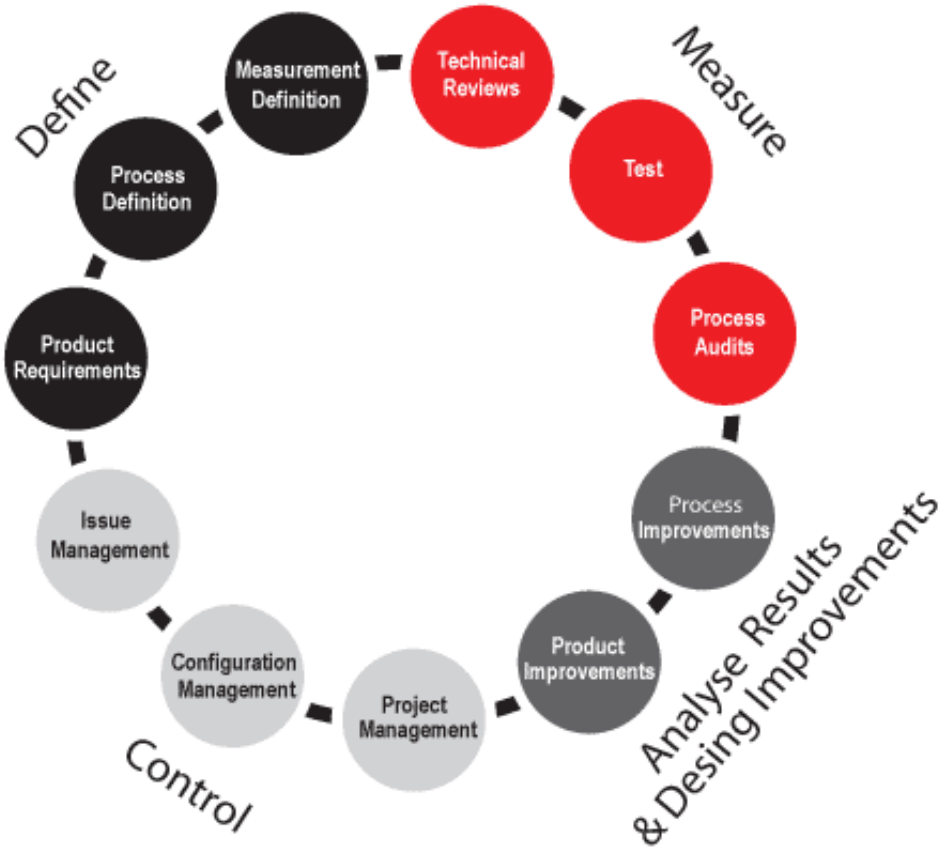
- Základ pro Business Continuity Plan (BCP).
  - Jak zajistit bezproblémový chod.
  - Omezit výpadky.
- Nešlo by to využít i pro něco jiného?
  - Děláme věci dobře?
  - Nemohli bychom to dělat lépe?
  - Jak to poznáme?
  - Lze podle něčeho srovnávat?
  - Co by nám pomohlo to zlepšit?
  - Co QUALITY MANAGEMENT.

Řízení kvality z pohledu IT

- Cíl - maximalizace dosahované kvality ve všech procesech životního cyklu – při návrhu, vývoji, implementaci, testování, dokumentaci, nasazování řešení, implementaci služeb, technické podpoře atd.
- *Quality Assurance* staví na tzv. **DMAIC** modelu - *Define, Measure, Analyse, Improve, Control*, který stanovuje pro procesy i produkty:
  - **Define:** definici
  - **Measure:** měření
  - **Analyse:** analýza výsledků měření
  - **Improve:** návrh zlepšení
  - **Control:** řízení zlepšování

http://www.adastra.cz/804_quality-assurance.aspx, *http://www.adastra.cz/792_testovani.aspx*

IT specifické procesy



- **Softwarový vývoj** (včetně důsledné implementace procesů testování, validace a verifikace), snižuje rizika spojená s tvorbou a dodávkou produktu.
- **Technická podpora** (včetně reakce na chyby v SW, vydávání opravných balíků, podpory uživatelů, reklamačního řízení).
- **Optimalizace správy ICT** (včetně procesů správy infrastruktury, uživatelských stanic, aplikací).

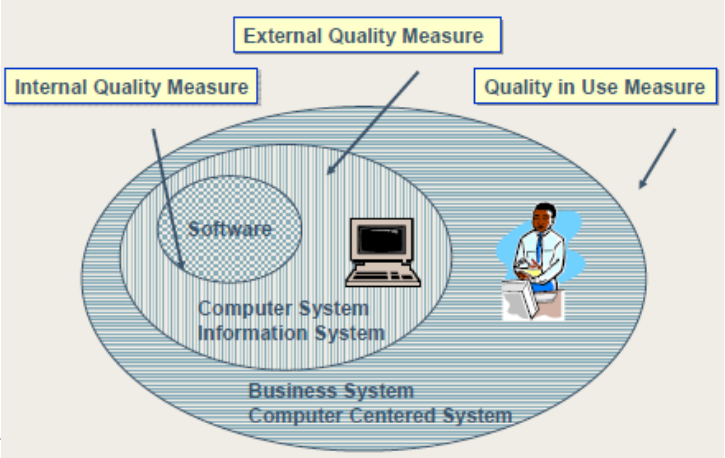
http://www.adastra.cz/804_quality-assurance.aspx, http://www.adastra.cz/792_testovani.aspx

Vybrané standardy pro vývoj SW

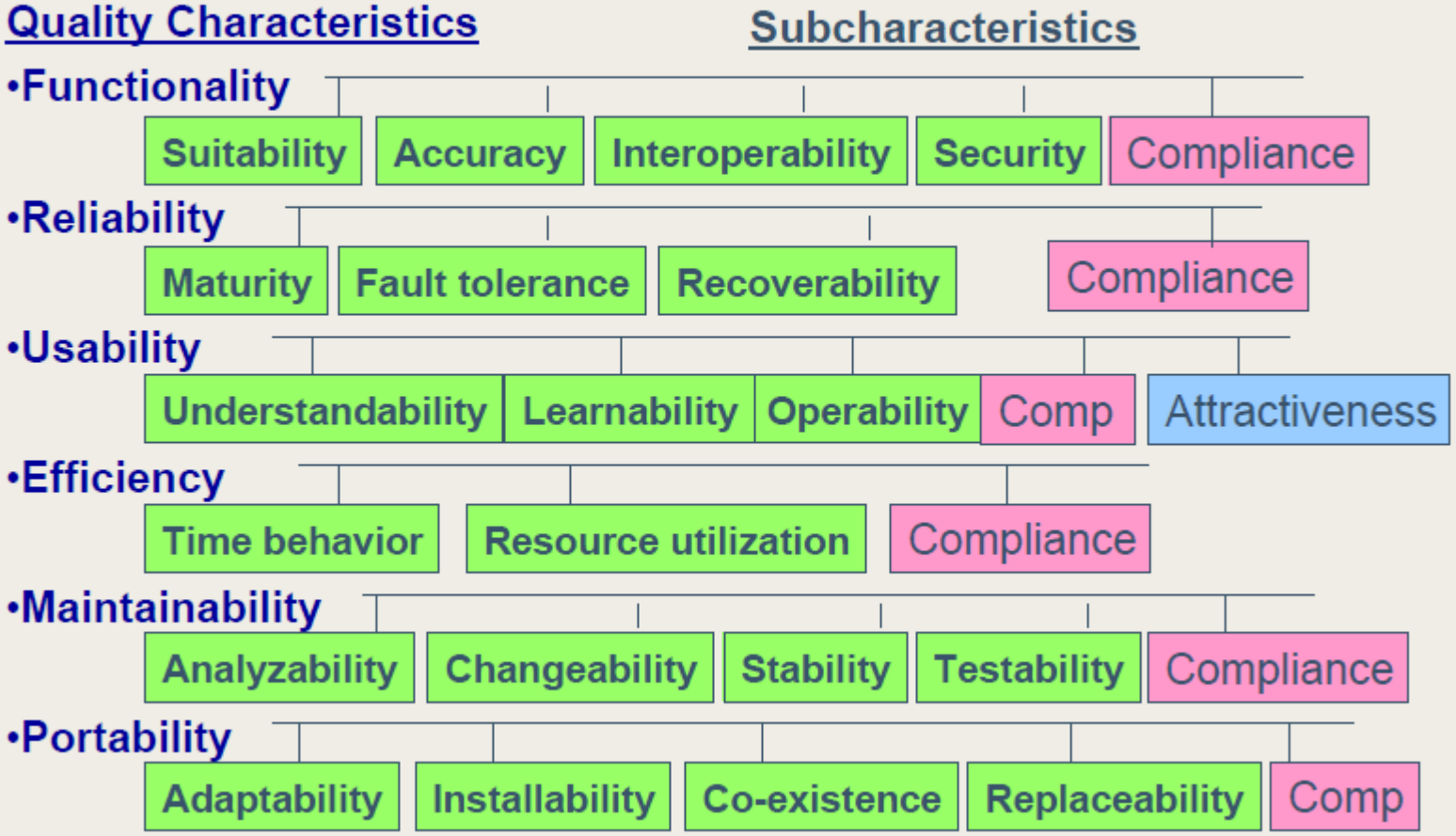
- **ISO 9000 (revize 1994, 2000, 2008)**
  - norma ISO, EU, ČR.
  - výrobní sféra i služby, obecné požadavky na organizaci.
- **CMM (Capability Maturity Model), CMMI (CMM Integration)**
  - Model vznikl z potřeby hodnotit pro ministerstvo obrany USA softwarové firmy při výběrových řízeních na státní zakázky v počátku osmdesátých let.
  - Software Engineering Institute, Carnegie Mellon University, USA (standardizace od 1987).
  - CMM - Capability Maturity Model (1990).
  - 1995 - verze modelu pro návrh technologických celků - SE-CMM (System Engineering CMM). Původní CMM označované nadále jako SW-CMM (softwarové CMM).
  - 2000 vznik CMMI (CMM Integrated), poslední revize v 2010 (v1.3), integrující standardy dohromady.
- **ISO 15504**
  - Software Process Improvement and Capability Determination (SPICE).
  - obdoba CMMI.
- **ISO 9126**
  - Standard pro hodnocení kvality sw (funkcionalita, spolehlivost, použitelnost, efektivita, udržovatelnost, přenositelnost (organizační, hw, sw).
  - adresuje obvyklé (problematické) momenty vývoje – změny priorit v průběhu projektu, cílů, apod.
- **ISO 27000**
  - řada standardů pro řízení bezpečnosti informací.

Požadavky na systémy dle ISO 9126 a ISO 25000

- Normy řady ISO 9126 stanovují obecné požadavky na jakost softwarového produktu. V současnosti jsou postupně nahrazovány novou (velmi podobnou) řadou norem ISO 25000, vyvíjených v rámci mezinárodního normalizačního projektu *SQuaRE*.
  - Norma ISO 9126-1 přináší model kvality a specifikuje 6 charakteristik jakosti, kde každá charakteristika má několik podcharakteristik
  - Normy ISO 9126-2 a ISO 9126-3 uvádí externí a interní metriky pro měření těchto charakteristik a
  - ISO/IEC 9126-4 se zabývá problematikou jakosti při použití softwarového produktu (Quality in Use).
- Stanovením a upřesněním priorit požadavků a následně konverzí abstraktních priorit řešení do měřitelných atributů podporuje standard jasný a srozumitelný pohled na záměry a cíle projektů.
- Standardizace umožňuje objektivizovat hodnocení opakovaně využitelných modulů, komponent, knihoven atd., včetně tzv. COTS/OTS (commercial off-the-shelf) sw, hw a technologických řešení.



ISO/IEC 9126-1 - Quality Model



Charakteristika a podcharakteristiky

- Funkčnost (Functionality):
  - Funkční přiměřenost (Suitability);
  - Přesnost (Accuracy);
  - Schopnost spolupráce (Interoperability);
  - Bezpečnost (Security);
  - Shoda ve funkčnosti (Functionality Compliance);
- Bezporuchovost (Reliability):
  - Zralost (Maturity);
  - Odolnost vůči vadám (Fault Tolerance);
  - Schopnost zotavení (Recoverability);
  - Shoda v bezporuchovosti (Reliability Compliance);
- Použitelnost (Usability):
  - Srozumitelnost (Understandability);
  - Naučitelnost (Learnability);
  - Provozovatelnost (Operability);
  - Atraktivnost (Attractivness);
  - Shoda v použitelnosti (Usability Compliance);
- Účinnost (Efficiency):
  - Časové chování (Time Behaviour);
  - Využití zdrojů (Resource Utilisation);
  - Shoda v účinnosti (Efficiency Compliance);
- Udržovatelnost (Maintainability):
  - Analyzovatelnost (Analysability);
  - Měnitelnost (Changeability);
  - Stabilnost (Stability);
  - Testovatelnost (Testability);
  - Shoda v udržovatelnosti (Maintainability Compliance);
- Přenositelnost (Portability):
  - Přizpůsobitelnost (Adaptability);
  - Instalovatelnost (Installability);
  - Slučitelnost (Co-existence);
  - Nahraditelnost (Replaceability);
  - Shoda v přenositelnosti (Portability Compliance).

ISO 9126 vs. 25000

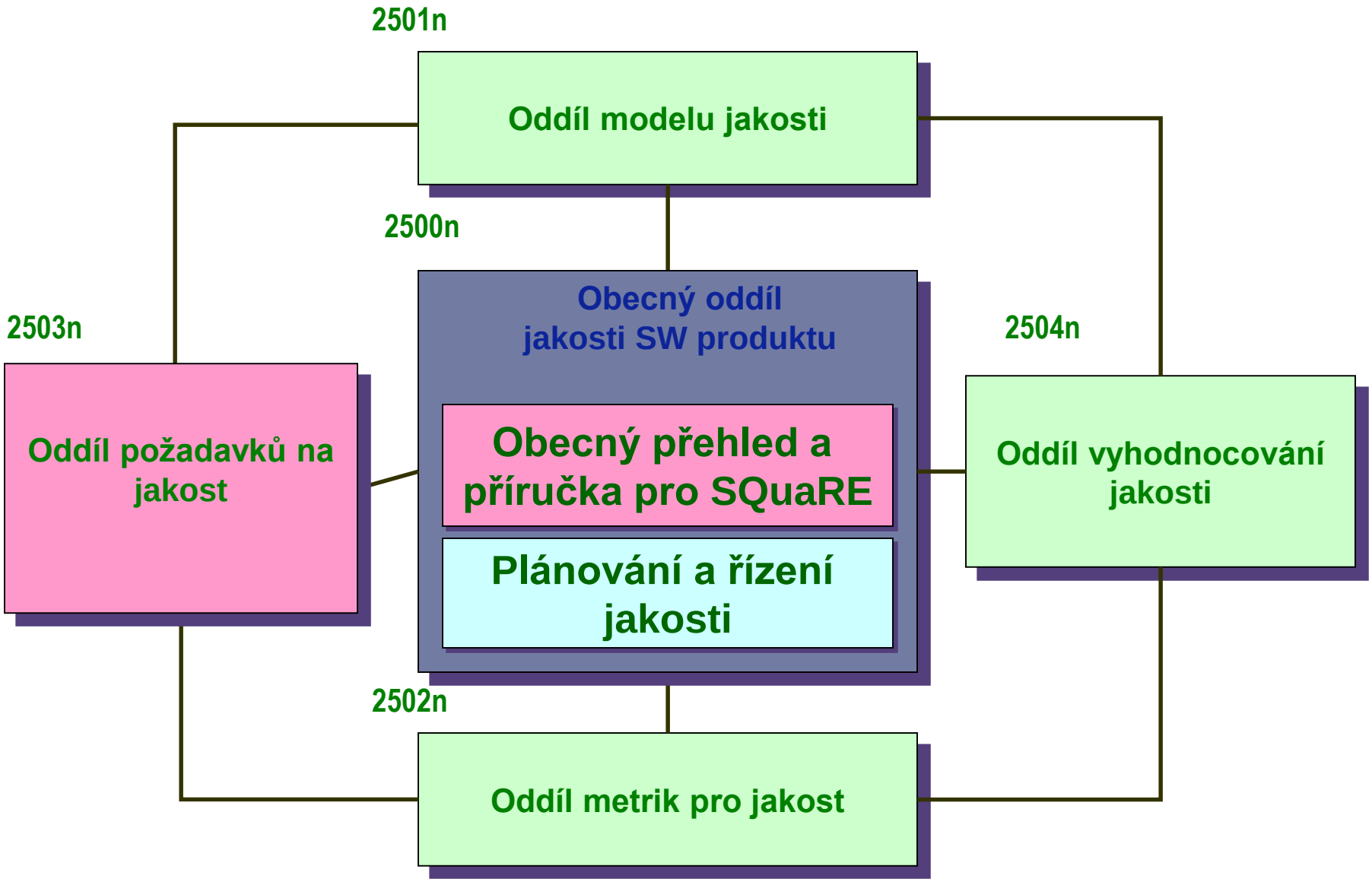
ISO 9126

- Metrik je navrženo příliš mnoho (přes 200), není jasné, které kdy vybrat
- Není jasné, jak formulovat potřeby a převést je do měřitelných požadavků
- Není jasné, kterou „jakost“ zkoumat
  - vnitřní (prediktory jakosti)
  - vnější (jakost produktu)
  - nebo jakost užití produktu (včetně „jakosti uživatele)
- 9126-1 Model jakosti
- 9126-2 Vnější metriky
- 9126-3 Vnitřní metriky
- 9126-4 Metriky pro jakost použití
- 9126-5 Základní softwarové metriky

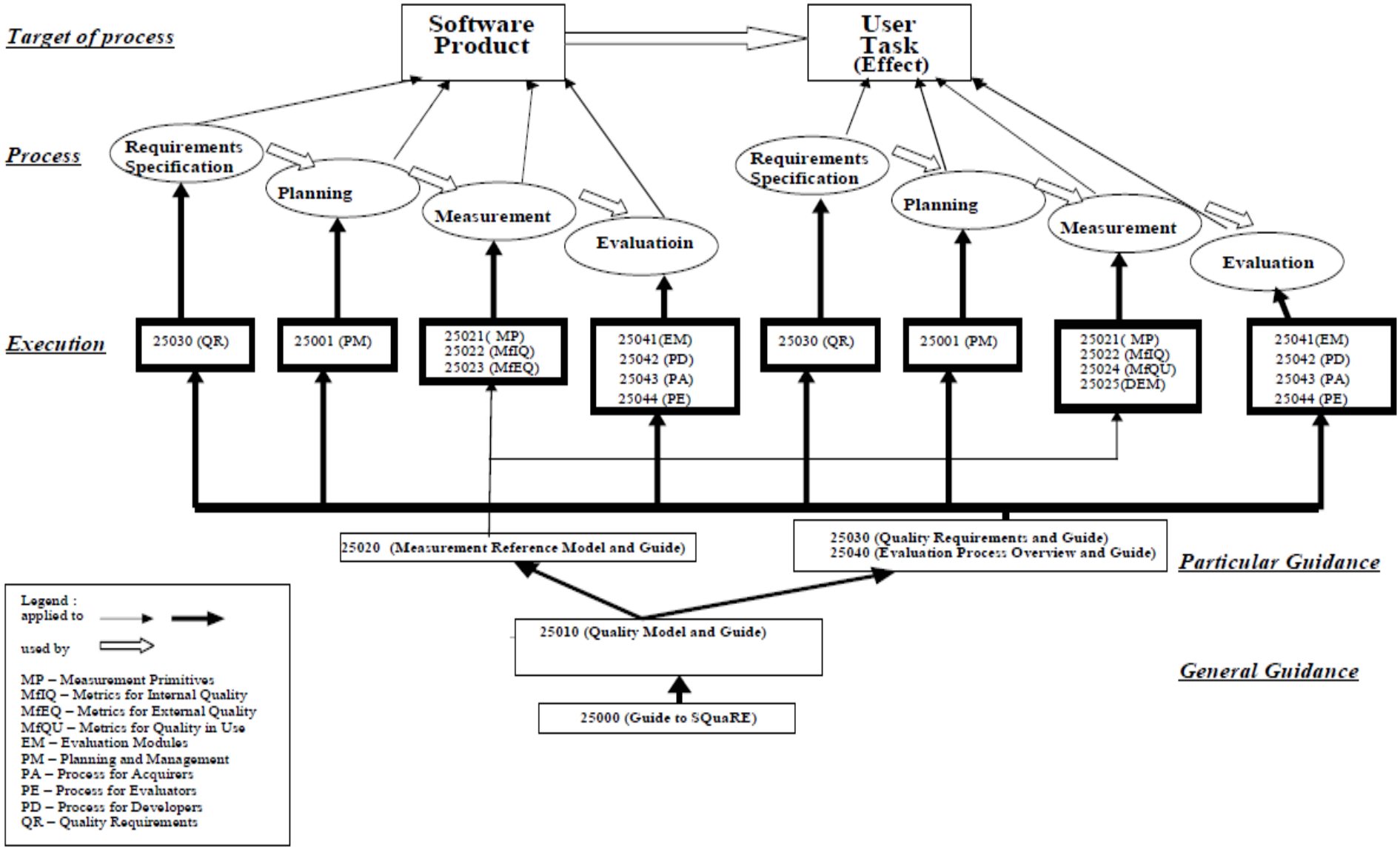
ISO 25000 (SQuaRE)

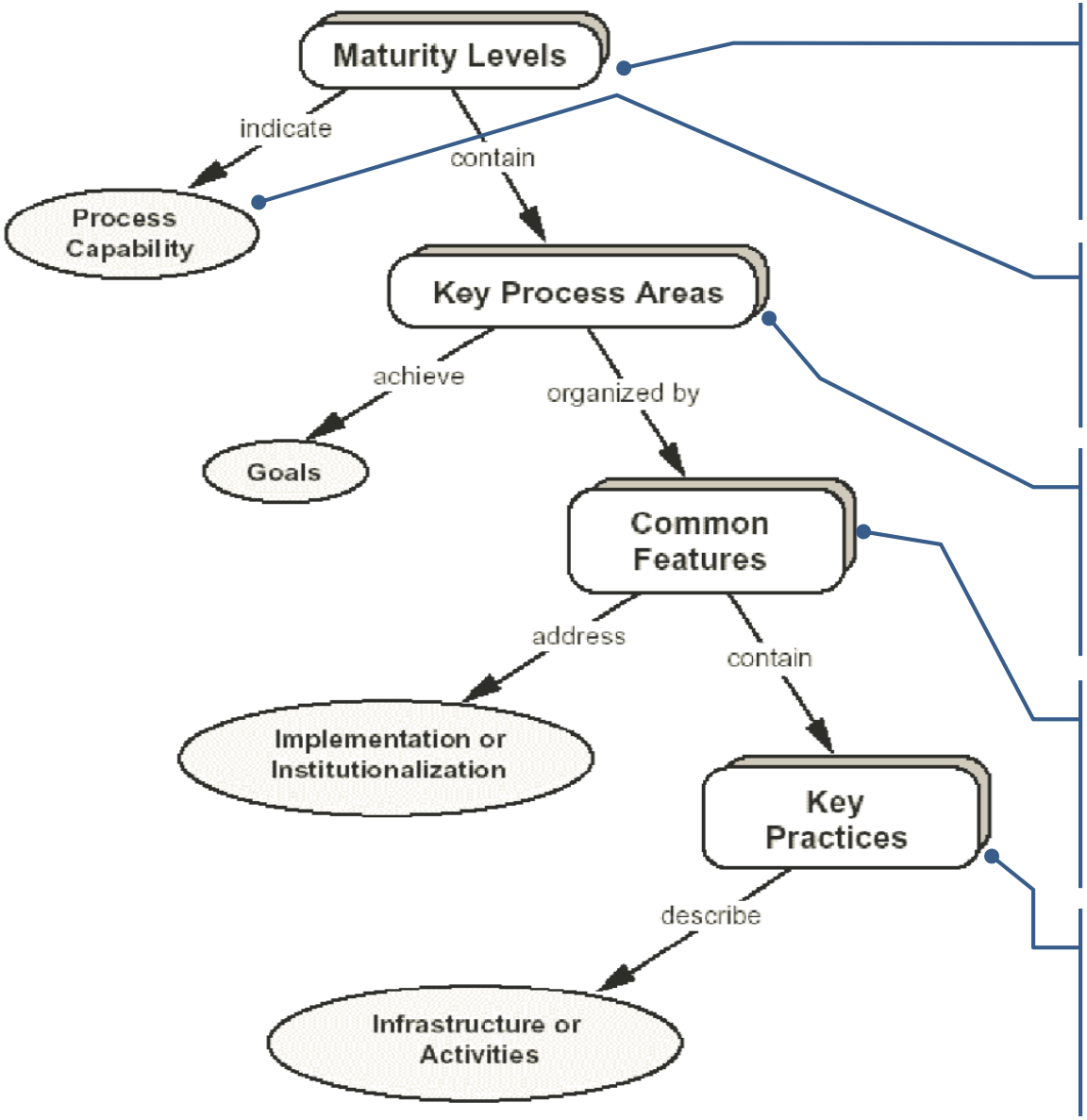
- Vytváří jednotnou architekturu řady norem a zastřešující příručku
- Vytvořit příručku pro to, jak užívat metriky
- Definuje primitiva pro měření, např. prvky měřené přímo (čas, počet, kategorie)
- Zavádí metriky pro objektivizaci požadavků na jakost
- 25010 Model jakosti
- 25020 Metriky
- 25030 Požadavky na jakost
- 25040 Vyhodnocování jakosti

SQuaRE architektura



SQuaRE general reference model





úrovně vyzrálosti procesu (*maturity levels)* = míra stability (v rámci projektu, mezi projekty), schopnosti detekce a opravy chyb, efektivity, predikovatelnosti výsledků

způsobilost (*capability)* = co je možné od organizace čekat v oblasti kvality

klíčové oblasti (*key process areas)* = *na co je* třeba se zaměřit pro další zkvalitnění procesu

společné rysy (*common features) zahrnují :* commitment to Perform, Ability to Perform, Activities Performed, Measurement and Analysis, and Verifying Implementation

klíčové techniky (*key practices) dávají návod jak* toho dosáhnout

Maturity Level	Charakteristika
Optimized (optimalizovaná) **- 5 -**	Zpětná vazba ovlivňuje následné fir… neustále zlepšovaly a dosahovaly předem defi… ních parametrů. Firma dosahuje trvale špičkové kvalit… ady na kvalitu softwaru enormě zvyšovaly.
Managed and Measurable (řízená) **- 4 -**	Firma má všechny procesy jasně definová… veny metriky, kterými vyhodnocuje jejich podíl a efekti… rocesů jsou postupně upravovány tak, aby se firma přiz… se podmínkám trhu, aniž by to mělo dopad na kvalitu vyvíjeného sw… kvalitativní parametry dosahují vysoké úrovně.
Defined process (definovaná) **- 3 -**	Software je vyvíjen podle předem stanoveného pos… etodicky, plánovitě, s využitím pokročilého projektového říz… vypracování požadovaného software v če… ními náklady a disponibilními zdroji. Provádí se pra… ání odchylek od plánu a přijímají se opatření. O kvalitu produk… lze se explicitně usiluje, proto je kvalita softwaru dodržována na velmi dobré úrovni a má tendenci vykazovat určité zlepšování.
Repeatable and intuitive (opakovatelná) **- 2 -**	Opakovaně se dosahuje dobrých výsledků… e postupů projektového řízení, ale z projektu na projekt se přená… pešné prvky řízení. Nicméně intuitivně zaběhané p… ání, že je potřeba pracovat kvalitně, vytvářejí dost stabilní prostředí… udržení přijatelné úrovně jakosti softwarových produktů.
Initial / ad hoc (úvodní) **- 1 -**	Dominují ad hoc procesy, sw je vytvářen bez proces… pravidel. Úspěch je spíše náhodný a závisí na individuálních schop… ny nejsou většinou dodržovány a ukončení vývoj… ejména úsilím na konci projektu. Celkové náklady na p… dodatečně dle nutných výdajů. Kvalita se systematicky nezajišťuje.

Kontinuálně zlepšovaný proces

Predikovatelný proces

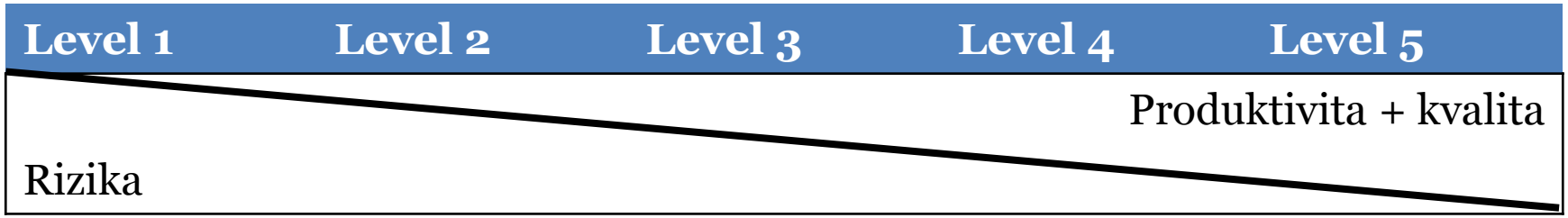
Standardní a konzistentní proces

Disciplinovaný proces

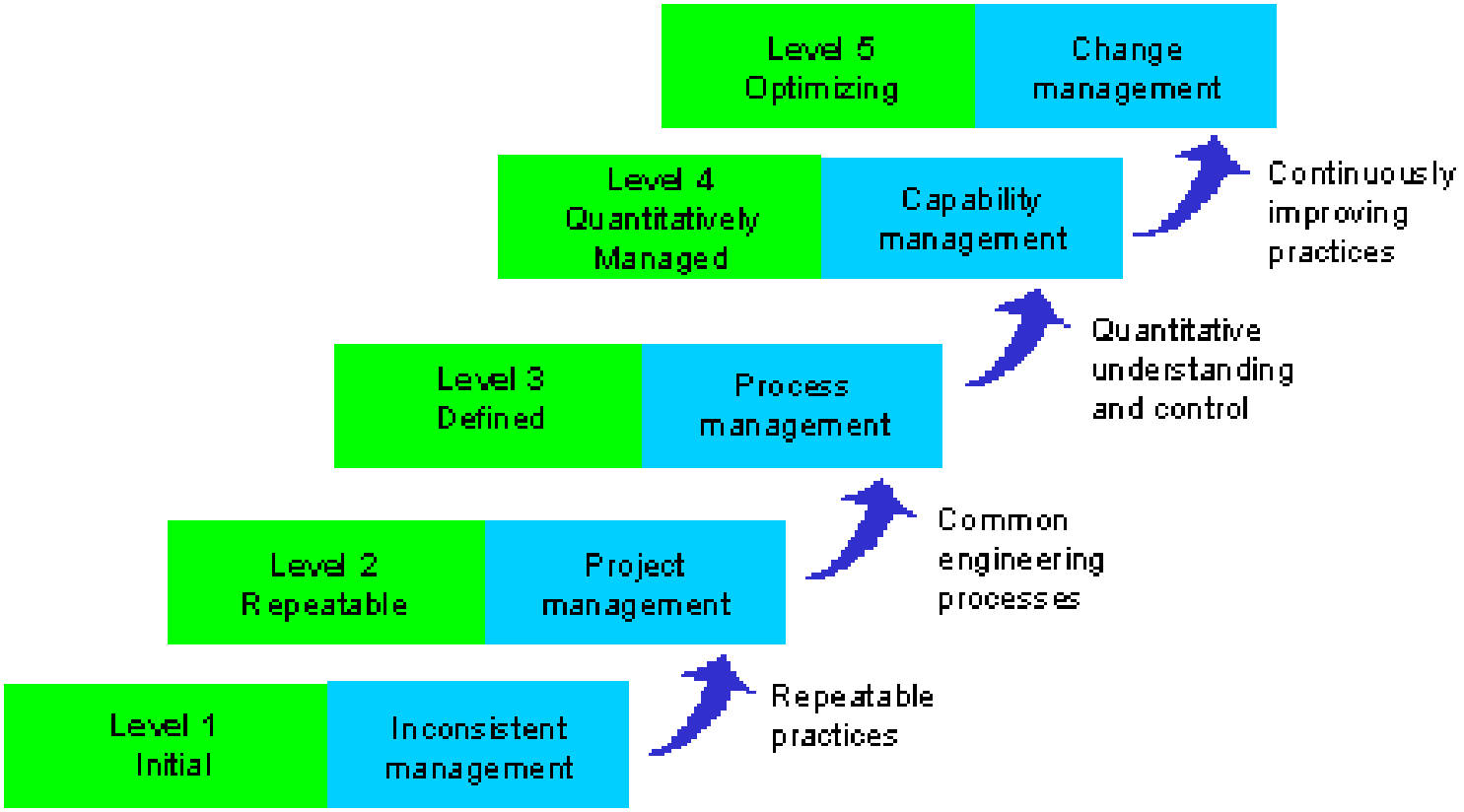
Nepredikovatelný proces

Klíčové aspekty úspěchu (Key Process Areas)

Level 2	Level 3	Level 4	Level 5
řízení požadavků	organizace firmy	kvantifikace procesů	prevence chyb
řízení projektů	organizace a definování procesů	řízení kvality SW	řízení změn
řízení nákupů komponent	zvyšování znalostí pracovníků	řízení konfigurace	optimalizační metody
využívání metod	integrace SW		business process reengineering



Maturity Levels of CMM



http://www.gartner.com/4_decision_tools/measurement/measure_it_articles/2003_0424/desrcibing_cmm.jsp

CMM jako univerzální model

- Další specializované modely:
  - **PM-CMM** (project management capability maturity model) - pro oblast projektového řízení.
  - **SW-TMM** (software testing maturity model) – pro oblast testování.
  - **P-CMM** (people capability maturity model) – pro oblast práce s lidskými zdroji.
  - **SA-CMM** (software aquisition capability maturity model) – pro oblast nákupu softwaru.
- Model CMM je nejvíce rozšířen v USA.
- Kritici namítají, že model není založen na teoretické bázi a existuje pro něj pouze vágní empirická podpora.

Rozšíření CMM o negativní úrovně I.

- **Úroveň 0 (negligent)** - Nedbalost, nepozornost, řada chybně navržených procesů a špatná až chaotická organizace firmy způsobuje, že vytvořený software má velký počet chyb, které se ani nestačí identifikovat, natož opravit. Termíny se nedaří plnit, plánované náklady se překračují, často se žádné plánování nákladů ani neprovádí. Práce, která je konec konců jaksi provedena, je nakonec zmařena v důsledku různých jiných nedostatků. Často vedení firmy i pracovníci spoléhají a očekávají nějaká zázračná řešení, která způsobí okamžité divy. Produkty se firmě od zákazníků neustále vracejí, aby byly opraveny a dopracovány, přičemž řada zákazníků žádá slevy na dodané produkty.
- **Úroveň –1 (obstructive)** – Řada kontraproduktivních opatření ve firmě a protichůdných procesů téměř znemožňuje vytvořit kvalitní software. Často se objevují odmítavá stanoviska k zavádění takových věcí jako: projektové řízení, řízení jakosti, uznávané metody tvorby software, produkty CASE apod., s odůvodněním, že se jedná jen o byrokratická, administrativní opatření, která komplikují programátorům a dalším zaměstnancům práci. Jedna reorganizace stíhá druhou, stejně zmatenou jako byla ta předchozí. Kvalita software je tak špatná, že zákazníci neustále produkty reklamují, firmu penalizují a postupně přecházejí k jiným firmám.

http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/immaturity.pdf

Rozšíření CMM o negativní úrovně II.

- **Úroveň –2 (contemptuous)** – Ve firmě pracovníci přehlížejí jakákoliv doporučení a zásady softwarového inženýrství. Programování je prohlašováno za umění, takže jakékoliv snahy o zavádění pořádku a systému do vývoje software je označováno jako útok na uměleckou tvořivost a svobodu. Nejsou vedeny žádné údaje o postupu vývoje softwaru, často jsou takové údaje záměrně ničeny. Zákazníkům není naslouchánо a jsou naopak přesvědčováni, že produkty, které dostaly jsou ty nejlepší, a jejich nefunkčnost je zapříčiněna vlastní nízkou úrovní znalostí uživatelů v používání počítačů.
- **Úroveň –3 (undermining)** – Samorostlé názory pracovníků firmy na tvorbu software jsou zcela mimo chápání normálních lidí a podkopávají důvěru veřejnosti v softwarové inženýrství.

http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/immaturity.pdf

Immaturity model☺

LEVEL	CHARACTERISTIC	KEY PROBLEM AREAS
foolish	(negligent) failure to allow successful development process	software reuse
stupid	(obstructive) counter-productive process imposed	development environments repositories
lunatic	(contemptuous) disregard for good software engineering institutionalised	automatic programming

SENSE

IDIOCY

http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/immaturity.pdf

CMM-Integrated (CMMI)

- Capability Maturity Model® Integration (CMMI)
  - SEI CMU 2002; v1.1 (2006 v1.2, 2010 v1.3)
  - Lepší vazba na ostatní modely a standardy (minimalizace překryvů).
  - Orientace na oblasti a ne na funkce.
  - Širší sada nejlepších technik.
- Rozdělení na oblasti zájmu
  - Product and service development — **CMMI for Development (CMMI-DEV),**
  - Service establishment, management, and delivery — **CMMI for Services (CMMI-SVC),** and
  - Product and service acquisition — **CMMI for Acquisition (CMMI-ACQ).**
- Implementace
  - průběžná (po KPA) × postupná (po úrovních).
  - KPA orientovaná více na výsledné produkty.

CMMI - CMF

- V závislosti na aplikační oblasti standardu CMMI se liší typy využívaných procesů. Klíčové oblasti (Key process areas), které jsou zahrnuty ve standardech všech typů organizací, tvoří tzv. CMMI Model Framework (CMF).

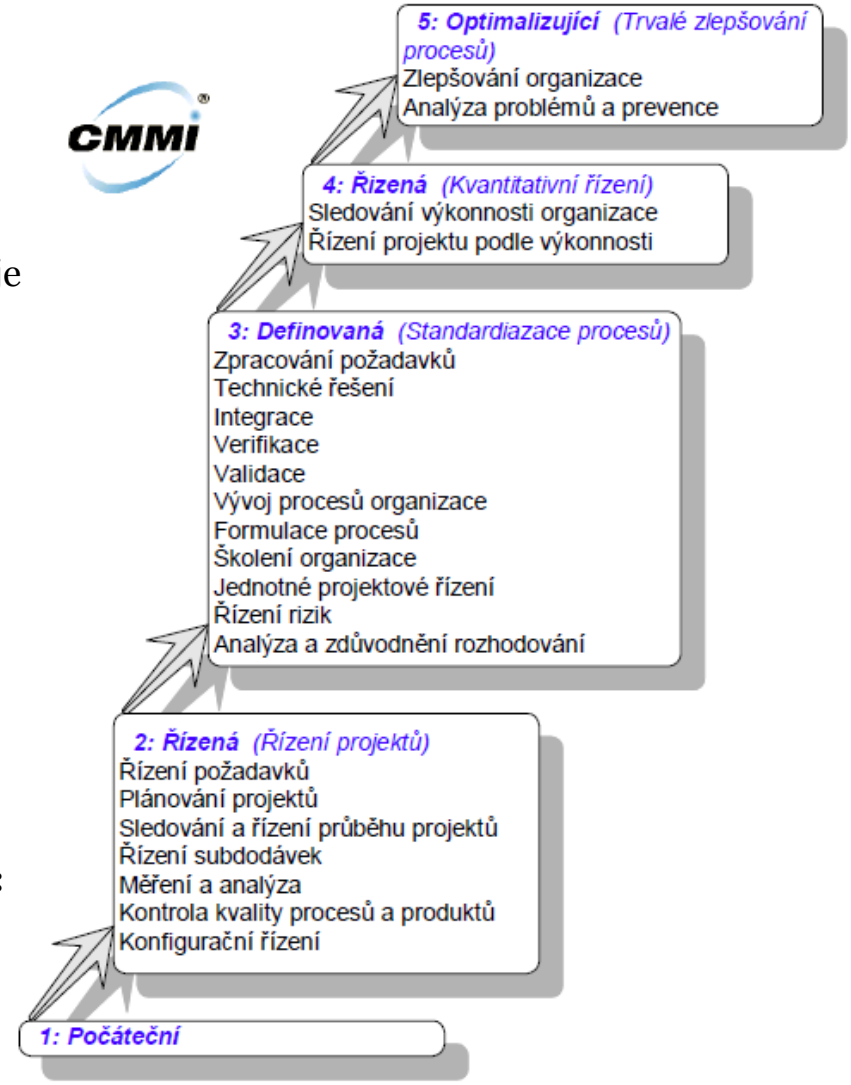
Abbreviation	Name	Area	Maturity Level
REQM	Requirements Management	Engineering	2
PMC	Project Monitoring and Control	Project Management	2
PP	Project Planning	Project Management	2
CM	Configuration Management	Support	2
MA	Measurement and Analysis	Support	2
PPQA	Process and Product Quality Assurance	Support	2
OPD	Organizational Process Definition	Process Management	3
CAR	Causal Analysis	Support	5

CMMI

Úrovně stupňovitého modelu CMMI

Stupňovitý model CMMI definuje 5 úrovní zralosti, model je navržen tak, aby firmy mohly kvalitu svých procesů přirozeně rozvíjet podle úrovní:

1. **Počáteční (Initial):** Týmy na této úrovni definované procesy nevykonávají nebo pouze částečně.
2. **Řízená (Managed):** Je stanoveno řízení projektů a činnosti jsou plánovány.
3. **Definovaná (Defined):** Postupy jsou definovány, dokumentovány a řízeny.
4. **Kvantitativně řízení (Quantitatively Managed):** Produkty i procesy jsou řízené kvantitativně.
5. **Optimalizující (Optimizing):** Tým soustavně optimalizuje své činnosti.



Struktura modelu CMMI *(Zdroj: školicí materiály DCIT)*

CMMI a ISO

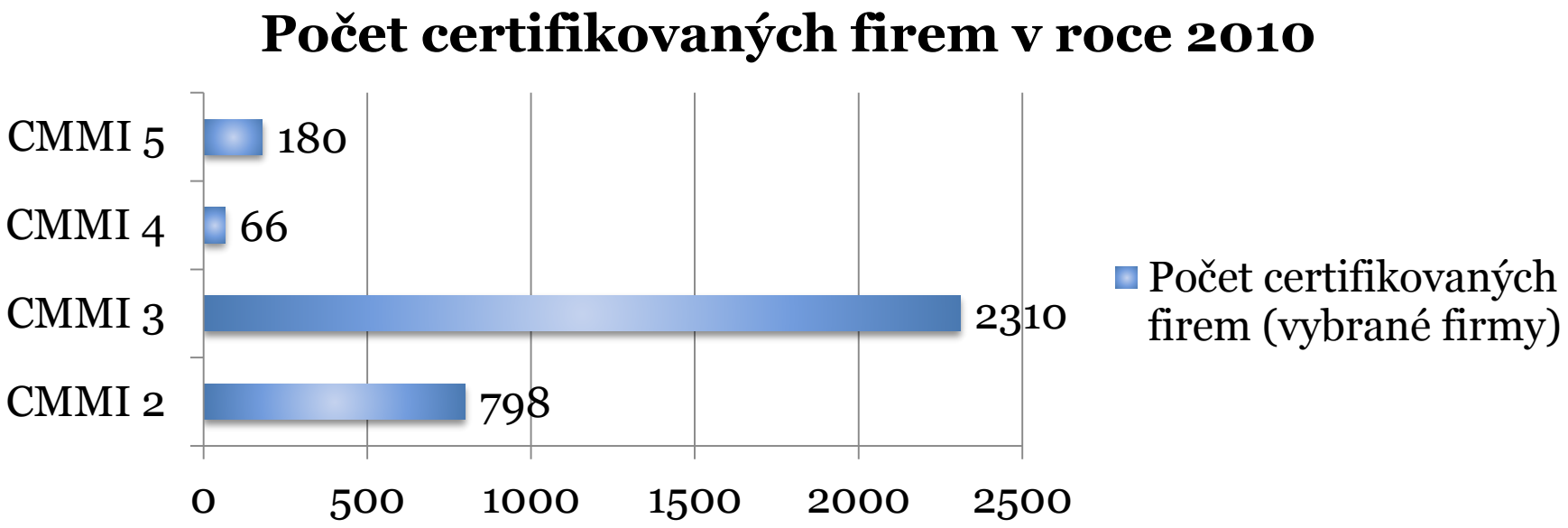
- CMMI se záměrně přizpůsobila standardu ISO 15504, protože globálně působící společnosti potřebují plnit oba standardy.
- Model je svým určením blízký dobře známému modelu ISO 9001, ale je mezi nimi několik zásadních rozdílů:
  - Standard ISO 9001 není určen pro žádnou konkrétní oblast a je aplikován na firmy z nejrůznějších oborů.
  - CMMI je určen pro vývojové týmy.
  - ISO 9001 je stručný standard, který definuje pouze cíle.
  - CMMI je podrobný model, který jde do podrobností, když definuje očekávané činnosti, jejich pracovní výstupy a definuje stupně zralosti.
  - CMMI má návodný charakter, takže je na jeho základě možné procesy přímo definovat.

CMMI není metodika!

- Na rozdíl od CMM necertifikuje úrovně, ale uděluje rating v kategorii 1 – 5.
- Metodika popisuje proces i postupy.
- Činnosti je možné ihned realizovat.
- Příklad metodik: ITIL, PMBOK, Prince 2, IPMA…
- CMMI je soubor cílů, které musí firma splnit.
  - CMMI nedefinuje, jak je naplnit.
  - Obdoba ISO 9000:2009.
  - Obsahuje nepovinná doporučení.
  - Často důvod, proč firmy implementují raději konkrétní metodiku.

Malá statistika CMMI

	2007	2008	2009	2010
Počet posouzení	1964	3113	4134	5499
Počet opakovaných posouzení	208	361	564	826
Podíl firem mimo USA	65%	69%	71%	74%
Počet zemí, kde jsou audit. spol..	59	63	67	69



http://sas.sei.cmu.edu/pars/pars.aspx

Zájem firem o CMMI

	2007	2008	2009	2010
Podíl komerčních certifikátů	70%	72%	74%	77%
Dodavatelé státu a armády	27%	23%	21%	19%
Podíl státních a armádních týmů	4,1%	5%	4,8%	4,5%

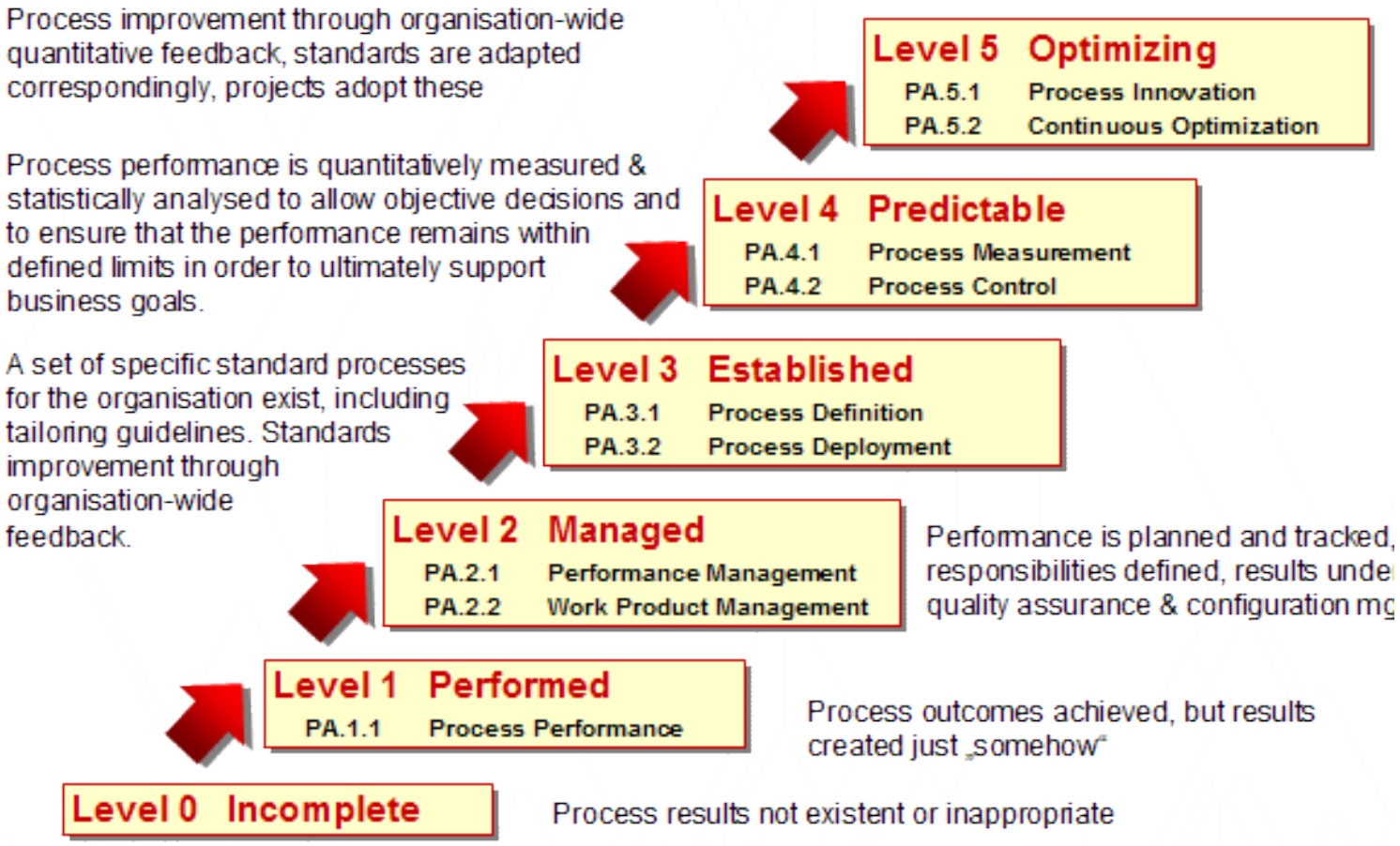
Velikost organizace	2007	2008	2009	2010
1 - 100	46%	51%	54%	58%
101 - 200	20%	20%	20%	18%
201 - 1000	27%	23%	21%	19%
1000+	7%	6%	5%	4%

http://www.pdqm.cz/Blog/CMMI-FAQ.html

SPICE

- Alternativní model jakostní tvorby software - např. ***ISO 15 504 - SPICE (Software Process Improvement and Capability dEterministic)***.
- Byl vytvořen se záměrem poskytnout rámec pro hodnocení softwarových procesů.
  - Byl odvozen od standardu CMM a ISO 12207 (standard životního cyklu sw).
  - Ani norma ISO 15 504 není metodickým návodem.
- Definuje referenční model prostřednictvím procesní dimenze a úrovně procesů.
- Rozeznává tři kategorie procesů:
  - Proces vztahu zákazník-dodavatel
  - Procesy softwarového inženýrství
  - Podpůrné procesy

Úrovně procesu

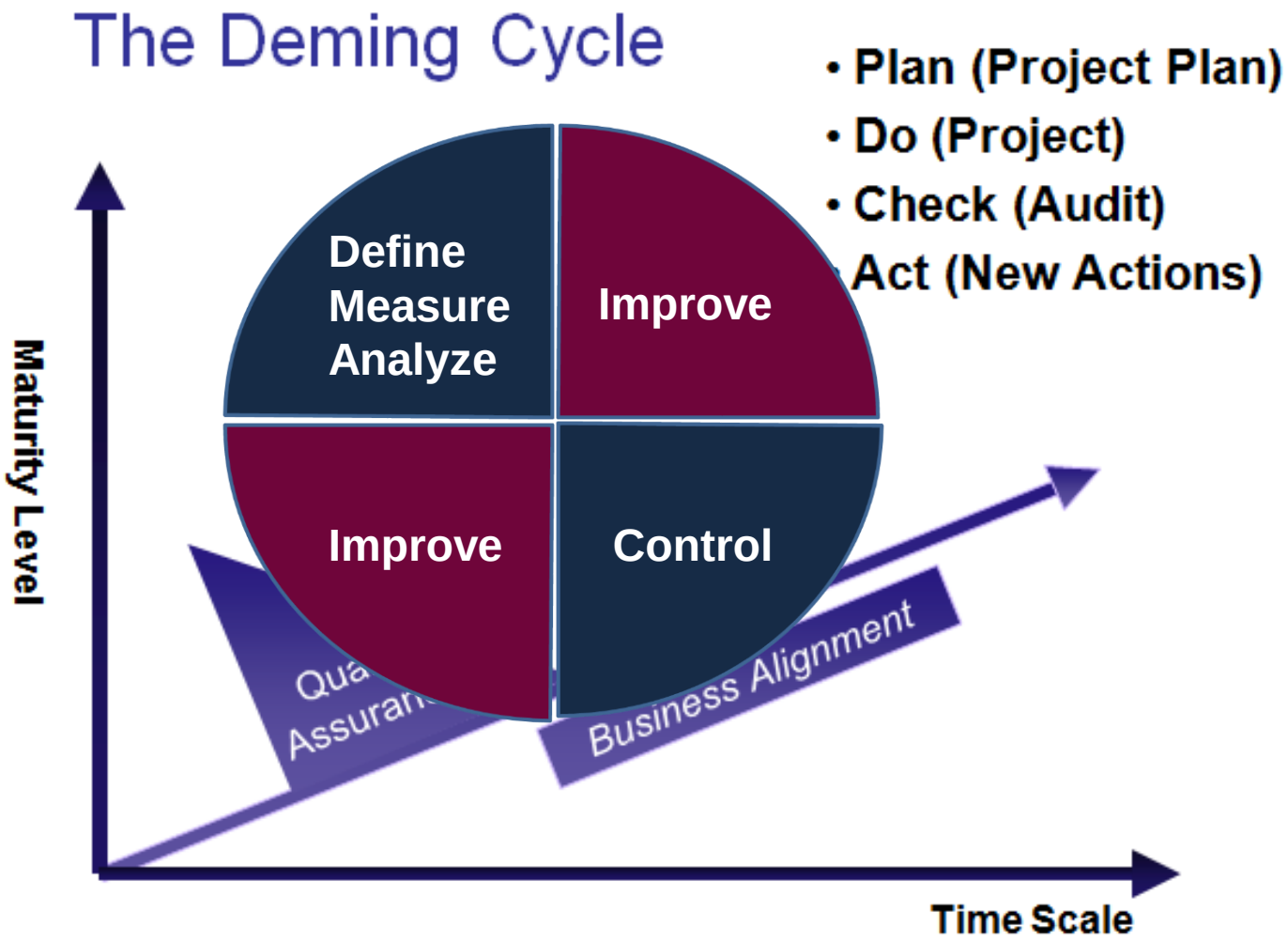


http://www.spice12drive.com/english/rel_services.htm

CMMI x SPICE

- Rozhodnutí, který model je vhodnější, je závislé především na zákaznících firmy a jejich požadavcích.
- Samotné modely jsou si podobné, liší se především úrovní detailu a zaměřením na určitou oblast podnikání.
- Pro firmy aspirující na dodávky do USA bude významnější model CMMI, evropské firmy zatím dávají přednost jiným standardům.
- Zájem o CMMI díky globální ekonomice a jeho dominantnímu postavení v Americe i Asii trvale roste i v Evropě.
- Pro automobilový průmysl je výhodnější používat odvozeného standardu Automotive SPICE .

Čím to začalo?



Závěr

Dotazy, připomínky, názory…